

Differential Evolution with an Evolution Path: A DEEP Evolutionary Algorithm

Yuan-Long Li, *Student Member, IEEE*, Zhi-Hui Zhan, *Member, IEEE*, Yue-Jiao Gong, *Student Member, IEEE*, Wei-Neng Chen, *Member, IEEE*, Jun Zhang, *Senior Member, IEEE*, and Yun Li, *Member, IEEE*

Abstract—Utilizing cumulative correlation information already existing in an evolutionary process, this paper proposes a predictive approach to the reproduction mechanism of new individuals for differential evolution (DE) algorithms. DE uses a distributed model (DM) to generate new individuals, which is relatively explorative, whilst evolution strategy (ES) uses a centralized model (CM) to generate offspring, which through adaptation retains a convergence momentum. This paper adopts a key feature in the CM of a covariance matrix adaptation ES, the cumulatively learned evolution path (EP), to formulate a new evolutionary algorithm (EA) framework, termed DEEP, standing for DE with an EP. Without mechanistically combining two CM and DM based algorithms together, the DEEP framework offers advantages of both a DM and a CM and hence substantially enhances performance. Under this architecture, a self-adaptation mechanism can be built inherently in a DEEP algorithm, easing the task of predetermining algorithm control parameters. Two DEEP variants are developed and illustrated in the paper. Experiments on the CEC'13 test suites and two practical problems demonstrate that the DEEP algorithms offer promising results, compared with the original DEs and other relevant state-of-the-art EAs.

Index Terms—Cumulative learning, differential evolution (DE), evolution path (EP), evolutionary computation.

I. INTRODUCTION

IN EVOLUTIONARY algorithms (EAs), two types of models are often used to generate new candidate solutions, which we refer to as a distributed model (DM) and a centralized model (CM) in this paper. For example, in the genetic algorithm (GA) [1], differential evolution (DE) [2], and particle swarm optimization (PSO) [3]–[9], each new candidate is

evolved from certain parental individuals directly. This can be regarded as following a DM because the parents are distributed among the population. On contrast, in evolution strategy (ES) algorithms, such as the estimation of distribution algorithm (EDA) [10] and the covariance matrix adaptation ES (CMA-ES) [11], all new individuals are generated using a centralized probability model. Similarly, in ant colony optimization (ACO) [12], [13], all ants use the same pheromone model to generate new individuals. Therefore, these algorithms can be regarded as using a CM to generate new candidates because the parental information is centralized with an intermediate model. Note that some other classification terms, such as explorative/exploitative, diversification/intensification, and individual-based/model-based, can also be used in classifying EAs from different perspectives. The term DM/CM is used in this paper because this paper focuses on the features of how parental information is utilized in the evolutionary process.

In this paper, we first analyse the different features and advantages of a DM and a CM, with a focus on combining their advantages in the development of new algorithms. In the literature, efforts are already seen in combining a CM with a DM [14]–[19]. In [14], for example, the use of some DM features to enhance the ACO algorithm is proposed, which shows some promising results. In [20], the teaching–learning optimization proposal combines a CM (a “teacher phase”) and a DM (a “learner phase”) for constrained optimization. Other relevant studies have focused on using a DM as a base algorithm, such as the combinations of DE and EDA [17], [19], PSO and EDA [15], and DE and CMA-ES [21]. These are however, based on a direct combination of two algorithms of a DM and of a CM. Hence, new populations are generated in both ways to introduce and maintain a higher diversity than using a single model alone.

However, such hybridizations result in complex reproduction processes and lack the advantage of efficient cumulative learning offered by a CM, such as the evolution path (EP) used in a CMA-ES. This paper therefore attempts to address these issues by developing a “DE with an EP” (DEEP) algorithm, using DE as the base algorithm with enhancement through an EP.

The remainder of this paper is organized as follows. In Section II, the DM and CM are fully analyzed, with reviews on several related search models. In Section III, the DEEP framework is introduced. In Section IV, two DEEP algorithms are developed and experiments on the CEC'13 test suites are performed to test the DEEP algorithms fully. The conclusion is drawn in Section V, with future work also highlighted.

Manuscript received March 10, 2014; revised June 29, 2014; accepted September 16, 2014. Date of publication October 9, 2014; date of current version August 14, 2015. This work was supported in part by the National High-Technology Research and Development Program (863 Program) of China under Grant 2013AA01A212, in part by the National Natural Science Foundation of China (NSFC) under Grant 61402545 and Grant 61379061, in part by the NSFC Key Program under Grant 61332002, and in part by the NSFC for Distinguished Young Scholars under Grant 61125205. This paper was recommended by Associate Editor P. N. Suganthan.

Y.-L. Li, Z.-H. Zhan, Y.-J. Gong, W.-N. Chen, and J. Zhang are with Sun Yat-sen University, Guangzhou 510275, China, also with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Guangzhou 510275, China, also with the Engineering Research Center of Supercomputing Engineering Software, Ministry of Education, Guangzhou 510275, China, and also with the Key Laboratory of Software Technology, Education Department of Guangdong Province, Guangzhou 510275, China (e-mail: zhanzh@mail.sysu.edu.cn).

Y. Li is with the School of Engineering, University of Glasgow, Glasgow G12 8LT, U.K.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCYB.2014.2360752

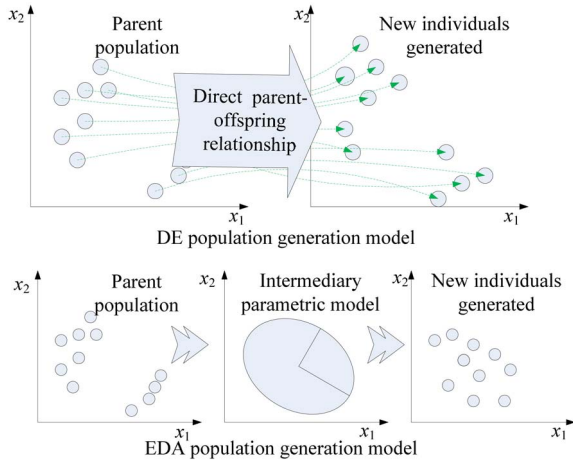


Fig. 1. Two kinds of population reproduction models: DE versus EDA.

II. REPRODUCTION MODELS

This section analyses the differences and advantages of the DM and CM models in generating new individuals. Take DE and EDA as examples for the use of DM and CM, respectively. Their differences are illustrated in Fig. 1.

Firstly, each new individual in the DE is generated based on its own parents. In this sense, different new individuals can be regarded to be generated according to the different models determined by their corresponding parental individuals. Therefore, the whole population in DE can be regarded as generated by combining multiple models. However, in an EDA, a uniform reproduction model is first created by using the information from some selected parents, and then the whole population is generated from a uniform Gaussian model. From this viewpoint, the DE reproduction model is regarded as a DM while the EDA a CM.

Secondly, in DE, the new population is directly generated by the parent population while in EDA the population is first built into a Gaussian model, which is then used as an intermediary parametric model to generate the new population. Therefore, a DM in DE is nonparametric whilst a CM in EDA is parametric.

Moreover, these two different types of models have different advantages. The DM scheme enjoys flexibility in nonparametric modeling and distributed individual generation. There is a direct generational parent-offspring relationship between the two successive generations. New individuals can be generated alongside the best parental individuals regardless of their distribution. On the other hand, as the CM scheme builds a Gaussian model as an intermediate model to generate new individuals, the CM scheme usually cannot work well on multimodal problems when the population is separately distributed in different locales. However, the CM scheme enjoys its advantage of parametric modeling, which can cumulate historical information continuously during the evolutionary process. The continuous adaptation of the CM can help adjust the shape and location of the Gaussian model to gain a better landscape approximation. Compared with the continuously learning feature of the CM, the DM scheme can only use the information of the immediate past generation, which is regarded as “discrete” in such a sense. Therefore, the

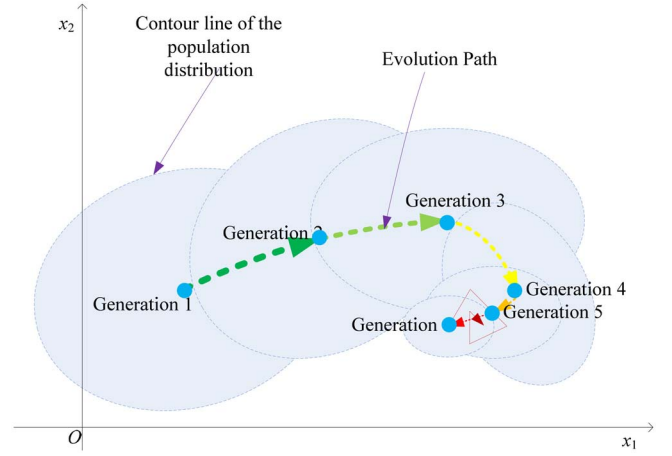


Fig. 2. Evolution path over a number of generations.

advantage in cumulative learning helps algorithms with a CM (such as the CMA-ES) work better on ill-shaped landscapes such as nonconvex or narrow corridor ones [11].

Given the analysis above, a question arises whether the advantages of both kinds of models can be combined. In the literature, some studies are reported in the sense of this question. Deb *et al.* [22] and Someya [23] assert that there are different kinds of crossover mechanisms in GA, such as the “parent mode crossover,” with which each offspring is generated by a corresponding parent, while “centric mode crossover” operations generate the whole offspring population by the center of the current population. Someya [23] proposes an asymmetric normal distribution crossover (ANDX) operation to tune these two crossover modes for improved performance. For ACO, Tsutsui [14] provides an interesting way to use the parental solutions to modify new solutions generated by the CM model with promising results.

Based on the analysis so far, we propose a new way other than direct combination of a DM and a CM. Here, the main structure of one model is retained while key features of the other are added to keep the hybridized algorithm as simple as possible. In this paper, we use the DE (which is based on a DM) as the base algorithm and combine some key features of a CMA-ES (which is based on a CM) into the DE framework. The cumulatively learned evolution path (EP) is used to adapt the parametric model of the CMA-ES for improved efficiency and performance. According to [11], the EP in the CMA-ES represents the migration path of the population center. Using EP to cumulatively modify the distribution shape of the Gaussian model plays a key role in the CMA-ES. As an illustration, Fig. 2 shows how the population center migrates, i.e., the EP.

Different from a superimposed DM and CM combination, all the individuals are still generated by the DM. In order to utilize the EP information, an additional differential vector is added to a new individual after it has been generated by the DE mutation and crossover operations. The added vector is generated by a Gaussian model along the direction of the EP, and therefore is efficient to guide the new individual to a promising region directed by the EP information.

However, sometimes the EP may detour from a specific center, like illustrated by the last several generations in Fig. 2. Nevertheless, the center of the detoured EP may indicate a good region for generating a new promising solution. Therefore, an anchor point which is a weighted mean of the recent population centers is used to maintain some momentum of the EP when it detours significantly. This way, the algorithm can be guided by both the EP direction and the anchor points.

Based on the analysis, it is apparent that the proposed DEEP scheme can be applied to more than one DE variant. In this paper, we illustrate and develop two DEEP algorithms, which are based on the conventional DE/rand variant and the well-performed adaptive DE variant, termed as JADE in [24]. In the remainder of this section, a brief review on DE and CMA-ES, which belong to DM and CM, respectively, are presented, along with a state-of-the-art DE/CMA-ES hybrid algorithm.

A. Models in DE

The DE algorithm is a typical EC algorithm with three basic operations for the population reproduction: mutation, crossover, and selection. In every generation, a population P first goes through mutation. The mutation operation of DE is very special in that it uses a linear combination of a base vector and one differential vector or more to generate a mutated vector. For example, for every individual x_i ($i = 1, \dots, \text{population_size}$) in P , its mutated vector v_i is generated by

$$v_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) \quad (1)$$

where r_1 , r_2 , and r_3 are three randomly selected individuals, and are also different from i . In this mutation scheme, the difference between individuals r_2 and r_3 is used as the mutation step while factor F controls the step scale. After a mutated population V is created, it will go through a crossover process with the parent population P . The crossover operation for DE can be in binary or exponential. Here, without loss of generality, we only illustrate the binary crossover, as it is used in most cases. The crossover operation recombines every pair of individuals of (v_i, x_i) to generate a new individual u_i as shown in

$$u_{ij} = \begin{cases} v_{ij}, & \text{if } \text{rand}(0, 1) \leq CR || j == jrand \\ x_{ij}, & \text{otherwise} \end{cases}, \text{ for } j = 1, \dots, D \quad (2)$$

where $\text{rand}(0,1)$ is a random number uniformly distributed within interval $[0, 1]$, D is the number of dimensions, CR is the crossover rate which controls how many dimensions of the newly generated individual are from the mutated vector v_i , and $jrand$ is a randomly selected index to make sure that at least 1-D of the mutated vector will enter into the newly generated individual.

The crossover process creates a temporary population U , which is evaluated and then enters into the selection procedure. This procedure uses a pair-wise comparison of U and P . As shown in the following equation, individual u_i and x_i are compared and the better one will enter into the next generation:

$$x_i^{\text{new}} = \begin{cases} x_i, & \text{if fitness}(x_i) \text{ is better than fitness}(u_i) \\ u_i, & \text{otherwise.} \end{cases} \quad (3)$$

The above basic framework of DE shows a very simple algorithm for continuous optimization problems. A recent survey on DE can be found in [25]. Recent studies on parameter adaptation [26]–[29] and parameter composition [30] have shown that algorithm parameters of DE are important for the DE to perform well on different kinds of problems. This observation has led to the use of different mutation strategies to improve the performance of DE [26], [27], [30]–[32]. Mallipeddi and Suganthan [33] have proposed the framework of using ensemble-based strategies and parameters. Resultant ensemble-based algorithms are shown to adaptively choose better performing strategies and parameters in dealing with various kinds of problems. In [34], a bare-bone algorithm is proposed to solve parameter setting problem. Moreover, multiple or different individuals are used to guide the search of DE in [24] and [35]–[37], while local or neighbor information is used to guide the search in [38]–[41]. Both methods are used to improve the efficiency while maintaining the global search ability of DE. Recently, theoretical study on the convergence characteristics of DE has also made important progress in [42]. DE based frameworks are also proposed for constrained optimization in [43] and dynamic optimization in [44] and [45]. Multistart JADE is proposed in [46], which as a restart method works well for CMA-ES and EDA [47] and could be further extended to other evolutionary computation methods such as DE. In [48], a DE is reported to offer efficient solutions to multimodal optimization problems. Overall, current DE studies mainly cover ensemble-based algorithms, which have shown advantage with automatic strategies and parameter settings; parameter adaptation methods, which are important to DE algorithm to solve different kinds of problems; and multiple guidance and local guidance, which have been shown very useful in improving the global search ability of DE. However, there has been no clear EP based studies for DE so far.

B. CMA-ES

The CMA-ES has proved to be one of the most promising EAs. The CMA-ES uses CMA to adapt the Gaussian distribution model to generate new individuals, the same as in a common EDA. In CMA-ES, a covariance matrix C represents the population distribution shape and is cumulatively updated.

Usually, there two ways to update the covariance matrix C , but test results reveals that the algorithm works well when C is updated by the “EP” only. The EP v_{EP} is defined as the cumulative migration step of the mean point of the population $(\bar{x}_g - \bar{x}_{g-1})$ between successive generations. The EP vector v_{EP} can modify the shape of the distribution and enlarge it along the direction of v_{EP} by modifying C . CMA-ES works well with such covariance matrix adaptation, which indicates that the EP information is very useful to improve the search efficiency. The cumulative EP information, as a feature of the whole population, is the key factor of the CM of CMA-ES. Such a CM is helpful for a DM based algorithm if properly used. More details of CMA-ES can be found in [11].

C. DCMA-EA

In [21], a hybrid algorithm of DE and CMA-ES, termed differential covariance matrix adaptation EA (DCMA-EA), is proposed. The DCMA-ES combines the DE operations into the CMA-ES new individual generation, which is aimed to enhance the global search ability of CMA-ES. The DCMA-EA improves the performance of CMA-ES on complex multimodal functions, but may not be efficient on some unimodal or simple multimodal problems. The DCMA-EA uses the CMA-ES algorithm as a base algorithm. It thus inherits all the complex parameters and computations such as eigen-decomposition used in CMA-ES. Hence, it is more complex than traditional DE variants.

III. DEEP FRAMEWORK

Following the analysis of key features in the DE and CMA-ES algorithms, one key feature is selected in this section to improve DE without needing eigen-decomposition or direct hybridization of DE and CMA-ES. This is to retain simplicity in implementation and to improve efficiency in operation.

A. Concepts, Problems, and Solutions of DEEP

The principle of DEEP is to use the EP learning model of CMA-ES to enhance the DE reproduction. In particular, DEEP incorporates an EP vector in the mutation of the original DE algorithm. The EP is computed by accumulating historical information of the population migration as an intermediate CM for the DE to predict direction and hence move faster toward a more promising region.

In developing the DEEP algorithm, the first issue to address is to assess whether or not the population has already gone too far in the EP direction and whether the population need to reverse the direction. Our solution to this problem is to use an adaptive parameter α to control the direction (either positive for forward or negative for backward) of the mutation and its step size. Parameter α thus lets the program decide the direction and step size.

The second issue is to decide whether the direction of the EP makes sense when the EP detours significantly. Observation reveals that the EP often detours round and round when the population is around a best region and slowly converges to the approximate center. Hence we can use the historical centers of the recent populations to calculate the centroid of their recent trajectory (i.e., an approximate center of the recent evolution path). The centroid can therefore be used as an anchor point c_{EP} to counter random detours.

B. Major Operations

Without loss of generality, EP is directly used as an additional mutation vector in DEEP rather than as a means of building a covariance matrix used in CMA-ES. This way, the framework of DEEP remains unchanged from the original DE algorithm.

At the beginning, a normal DE population is generated. For every subsequent generation in the evolution process, each individual in the generation goes through the DE mutation and

crossover operations as shown in (1) and (2). Then an additional EP mutation is added to the new individual u_i (for $i = 1, \dots, \text{population_size}$)

$$u'_i = u_i + F \cdot sCR \cdot (\alpha_i \cdot v_{EP} + \beta_i \cdot (c_{EP} - u_i)) \quad (4)$$

where F and $sCR = CR$ are the same parameters as F and CR in (1) and (2), v_{EP} is the cumulative learning EP vector and c_{EP} is the anchor point. Parameter α_i is generated with a Gaussian distribution to control the size and direction of EP and β_i is a parameter used to control the step size to the anchor point c_{EP} .

Based on (4), u_i is further guided by the EP directional information and the anchor point. It should be noted that v_{EP} is a vector along the population's promising EP direction and is therefore directly added to u_i , while c_{EP} advocates a promising position with the β_i -weighted difference ($c_{EP} - u_i$) "correcting" u_i .

The v_{EP} vector is the migration vector of the mean point of the population Center_g from the previous generation to the current generation g

$$v_{EP} = \text{Center}_g - \text{Center}_{g-1} \quad (5)$$

where Center_g is the mean position of the first s best individuals in the population in generation g , defined as

$$\text{Center}_g = \text{mean}(x_{\text{best}_1}, x_{\text{best}_2}, \dots, x_{\text{best}_s}). \quad (6)$$

The anchor point c_{EP} of the cumulative learning is defined as

$$c_{EP} = \lambda \cdot c_{EP} + (1 - \lambda) \cdot \text{Center}_g. \quad (7)$$

Based on (5) and (7), v_{EP} and c_{EP} collect the EP information from two different aspects. Firstly, v_{EP} is for a fast response to the recent population migration vector, which can be very helpful when a long range migration is needed. Secondly, c_{EP} is updated using a learning parameter λ , which can be taken as a weighted mean of the recent population centers. This is useful when the population experience a contraction process. A λ value of 0.5 appears to be consistently good based on our experimental studies (Section IV). For the control parameters α_i and β_i in (4), simple self-adaptation suffices.

With the EP applied to new individuals, DEEP reproduction benefits from advantages of both distributed and CMs. In particular, each individual inherits from its parents by (1) and (2), hence maintaining population diversity and good sampling of the search space. Then, (4) acts as a CM to track the best parents' moving direction in the generation of new individuals. Fig. 3 illustrates how a DEEP mutant is generated.

C. Parameter Adaptation in DEEP

In (4), parameter F is adopted to keep the EP with a similar scale to the original DE mutation, while sCR is to control the scale of EP information as it is added after the original DE mutation and crossover operations. Experimental results advise that using sCR to scale the EP information instead of applying it before the crossover is better, for this preserves the geometrical characteristics of the EP.

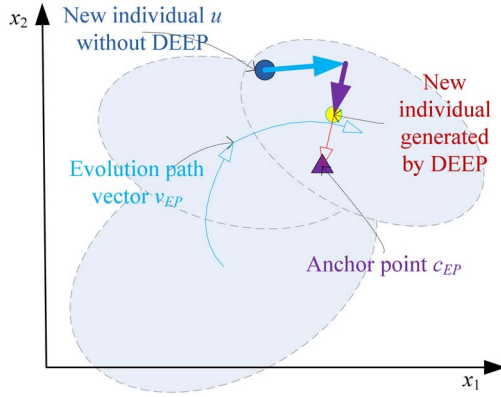


Fig. 3. DEEP generates a new individual by adding v_{EP} and a step toward c_{EP} .

Parameters α_i and β_i are self-adaptive in the algorithm. Parameter α_i is generated with a Gaussian distribution, where both the size and sign of α_i are used

$$\alpha_i = 2N(\alpha_m, \alpha_{sig}). \quad (8)$$

The distribution is centered at α_m , which creates a balance factor for the EP mutation direction. If $\alpha_m > 0$, the distribution gears toward the forward direction of v_{EP} . If $\alpha_m < 0$, the distribution gears toward the backward direction of v_{EP} . Parameter α_m is adaptively adjusted in a way similar to how CR adapts in JADE [24]. Because $N(\alpha_m, \alpha_{sig}) \in (-\infty, \infty)$, α_i is truncated into an interval $[-\alpha_{max}, \alpha_{max}]$ in implementations, as shown in

$$\alpha_i = \begin{cases} \alpha_{max}, & \text{if } \alpha_i > \alpha_{max} \\ -\alpha_{max}, & \text{if } \alpha_i < -\alpha_{max}. \end{cases} \quad (9)$$

After the evaluation of the population, those α_i values that have helped generate better offspring are used to tune α_m adaptively by way of

$$\alpha_m = 0.9 \alpha_m + 0.1 \text{mean}(\alpha_{good})/2 \quad (10)$$

where $\text{mean}(\alpha_{good})$ is the mean value of α_i that have helped the individuals generate better new individuals. The momentum from α_{good} helps the algorithm adapt itself in the distribution of the EP during the search process.

Similarly, a self-adaptive adjustment for the scale factor β_m is formulated. For each individual, β_i is generated by

$$\beta_i = \begin{cases} 0, & \text{if } \beta_i < 0 \\ \beta_{max}, & \text{if } \beta_i > \beta_{max} \\ N(\beta_m, \beta_{sig}), & \text{otherwise} \end{cases} \quad (11)$$

where β_m and β_{sig} are the center point and standard deviation of the Gaussian distribution used to generate β_i . Those β_i values, β_{good} , that have helped the individuals generate better new individuals are used for an adaptive center of the Gaussian distribution, as shown in

$$\beta_m = 0.9 \beta_m + 0.1 \text{mean}(\beta_{good}). \quad (12)$$

The self-adaptation of the control parameters α_i and β_i relieves the user from setting those parameters for the EP mutation. Overall, the framework of DEEP is shown in Fig. 4.

General DEEP Framework

Begin

1) **Initialization.** Besides the original DE parameters like the population, v_{EP} is set as zeros vector and c_{EP} is set as $Center_0$. α_m and β_m are set as 0.

2) Iteration loop:

For each individual i ($1 \leq i \leq \text{popsize}$) in the population

Generate α_i according to (9);

Compute β_i according to (11);

Do normal DE mutation and crossover to get u_i ;

Do EP mutation to each u_i according to (4);

Evaluate the new individuals;

Do normal DE selection;

Update α_m according to (10);

Update β_m according to (12);

Update v_{EP} according to (5) (6);

Update c_{EP} according to (7);

End of for

3) If stop condition met, go to 4); else go to 2).

4) Stop.

End

Fig. 4. DEEP framework.

TABLE I
PARAMETER SETTINGS USED IN THE TESTS

Independent run times	51 for CEC'13 test suite, 25 for practical problems
Population size	100 for DE/rand/1 and DE/best/2, 400 when $D=100$
F and CR	$F=0.5$, $CR=0.9$ for DE/rand/1 and DE/best/2
λ and s	0.5, 20
Initial (α_m, β_m)	(0, 0)
$(\alpha_{sig}, \beta_{sig})$	(0.3, 0.05)
$(\alpha_{max}, \beta_{max})$	(2, 0.25)
Maximum FEs	$D \times (1E+4)$, while $1.5E+5$ for practical problems

IV. TWO DEEP ALGORITHMS ILLUSTRATED WITH EXPERIMENTAL STUDIES

In this section, we illustrate the DEEP framework through two example DEEP algorithms and their experimental studies. We first apply DEEP to a canonical DE variant DE/rand/1 and to a state-of-the-art DE variant JADE [24] to develop a DE/rand/EP and a JADE with an EP (JADEEP), respectively. Experimental comparisons are then made with the original algorithms and other relevant EAs, including the DCMA-EA [21], CMA-ES [11], composite DE (CoDE) [30], self-adaptive DE (SaDE) [27], self-adaptive ensemble parameters and strategies DE (SaEPSDE) [49] (a recent version of EPSDE), and self-adaptive DE with neighborhood search (SaNSDE) [26]. In the experiments, all algorithms are tested on the single-objective box-constrained continuous optimization problems in the CEC'13 benchmarks [50] and on two practical problems [52]. Test codes of JADE, CoDE, and CMA-ES are provided in [30]. Test codes of SaEPSDE and SaNSDE are obtained from their authors, respectively. Test code of DCMA-EA is implemented according to the pseudo code provided by the authors.

A. Development of DEEP From DE/rand/1 and JADE

In the development of a DEEP algorithm from the traditional DE algorithm, DE/rand/1, all the settings in Section III are used. The bound handling method is adopted from JADE, as shown in

$$u_{ij}' = \begin{cases} (x_{ij} + \text{ubound}_j)/2, & \text{if } u_{ij}' > \text{ubound}_j \\ (x_{ij} + \text{lbound}_j)/2, & \text{if } u_{ij}' < \text{lbound}_j \end{cases} \quad (13)$$

TABLE II
TEST RESULTS OF DE/RAND/EP VERSUS DE/RAND/1 IN TEST PROBLEMS WITH $D = 30, 50$, AND 100 (NUMBER OF FES IS SHOWN IN PARENTHESES IF MEAN ERROR IS ZERO; BETTER MEAN ERRORS ARE SHOWN IN BOLDFACE; SIGNIFICANTLY DIFFERENT RESULTS ARE HIGHLIGHTED IN SHADE)

D	DE/rand/1	DE/rand/1/EP	D	DE/rand/1	DE/rand/1/EP	D	DE/rand/1	DE/rand/1/EP
30	Mean \pm std	Mean \pm std	50	Mean \pm std	Mean \pm std	100	Mean \pm std	Mean \pm std
1	0 (1.14E+05 \pm 2.38E+03)	0 (5.24E+04 \pm 7.47E+02)	1	0 (1.79E+05 \pm 3.58E+03)	0 (7.33E+04 \pm 1.62E+03)	1	1.26E+00 \pm 3.69E-01	0 (5.60E+05 \pm 4.24E+03)
2	4.02E+05 \pm 2.40E+05	1.24E+05 \pm 7.74E+04	2	3.05E+06 \pm 1.07E+06	3.53E+05 \pm 1.32E+05	2	7.04E+08 \pm 8.98E+07	6.61E+07 \pm 1.26E+07
3	2.84E+01 \pm 1.64E+02	4.38E+05 \pm 1.02E+06	3	1.80E+05 \pm 2.61E+05	2.97E+06 \pm 4.05E+06	3	3.21E+10 \pm 6.66E+09	2.98E+05 \pm 7.52E+05
4	1.05E+03 \pm 6.40E+02	4.56E+02 \pm 2.66E+02	4	1.95E+04 \pm 3.41E+03	2.05E+03 \pm 5.41E+02	4	1.86E+05 \pm 1.09E+04	9.73E+04 \pm 8.65E+03
5	0 (1.52E+05 \pm 3.20E+03)	0 (8.98E+04 \pm 2.38E+04)	5	0 (2.47E+05 \pm 6.41E+03)	0 (1.42E+05 \pm 1.38E+04)	5	2.49E+00 \pm 4.30E-01	0 (7.63E+05 \pm 1.58E+04)
6	9.48E+00 \pm 5.72E+00	2.68E+00 \pm 7.90E+00	6	4.36E+01 \pm 8.00E-01	4.34E+01 \pm 1.35E-13	6	2.98E+02 \pm 2.19E+01	1.92E+02 \pm 2.62E+01
7	2.12E-01 \pm 5.75E-01	2.91E-01 \pm 4.49E-01	7	1.80E+00 \pm 2.10E+00	4.81E+00 \pm 3.22E+00	7	1.21E+02 \pm 8.79E+00	1.93E+00 \pm 1.33E+00
8	2.10E+01 \pm 5.53E-02	2.09E+01 \pm 5.07E-02	8	2.11E+01 \pm 3.59E-02	2.11E+01 \pm 3.65E-02	8	2.13E+01 \pm 2.91E-02	2.13E+01 \pm 2.84E-02
9	2.63E+01 \pm 1.50E+01	9.16E+00 \pm 9.24E+00	9	6.74E+01 \pm 1.33E+01	2.40E+01 \pm 1.99E+01	9	1.60E+02 \pm 1.74E+00	1.59E+02 \pm 1.62E+00
10	6.96E-03 \pm 6.88E-03	6.22E-02 \pm 4.59E-02	10	3.06E-02 \pm 1.89E-02	1.06E-01 \pm 6.04E-02	10	1.36E+02 \pm 3.09E+01	4.71E-02 \pm 2.45E-02
11	1.38E+02 \pm 2.45E+01	1.16E+02 \pm 3.42E+01	11	2.05E+02 \pm 4.52E+01	4.10E+01 \pm 5.50E+01	11	8.71E+02 \pm 2.00E+01	7.74E+02 \pm 1.84E+01
12	1.79E+02 \pm 1.03E+01	1.62E+02 \pm 9.40E+00	12	3.55E+02 \pm 1.39E+01	3.25E+02 \pm 1.26E+01	12	9.13E+02 \pm 2.35E+01	8.09E+02 \pm 1.85E+01
13	1.80E+02 \pm 1.15E+01	1.61E+02 \pm 9.41E+00	13	3.55E+02 \pm 1.32E+01	3.22E+02 \pm 1.21E+01	13	9.08E+02 \pm 2.51E+01	8.04E+02 \pm 1.78E+01
14	6.28E+03 \pm 4.90E+02	6.69E+03 \pm 3.33E+02	14	1.16E+04 \pm 1.14E+03	1.27E+04 \pm 4.25E+02	14	3.01E+04 \pm 4.11E+02	3.01E+04 \pm 4.51E+02
15	7.04E+03 \pm 2.44E+02	7.07E+03 \pm 2.00E+02	15	1.38E+04 \pm 3.73E+02	1.38E+04 \pm 3.27E+02	15	3.04E+04 \pm 4.64E+02	3.01E+04 \pm 5.48E+02
16	2.47E+00 \pm 2.78E-01	2.44E+00 \pm 2.88E-01	16	3.30E+00 \pm 3.41E-01	3.30E+00 \pm 2.74E-01	16	3.99E+00 \pm 2.37E-01	4.00E+00 \pm 2.66E-01
17	1.87E+02 \pm 1.82E+01	1.87E+02 \pm 9.80E+00	17	3.30E+02 \pm 2.96E+01	3.71E+02 \pm 1.21E+01	17	9.85E+02 \pm 2.13E+01	8.92E+02 \pm 1.91E+01
18	2.14E+02 \pm 9.17E+00	1.87E+02 \pm 1.03E+01	18	4.01E+02 \pm 1.48E+01	3.72E+02 \pm 1.45E+01	18	1.00E+03 \pm 2.04E+01	8.97E+02 \pm 2.32E+01
19	1.51E+01 \pm 9.59E-01	1.33E+01 \pm 7.98E-01	19	2.97E+01 \pm 1.59E+00	2.46E+01 \pm 3.06E+00	19	8.11E+01 \pm 1.94E+00	6.97E+01 \pm 1.78E+00
20	1.22E+01 \pm 2.58E-01	1.15E+01 \pm 3.20E-01	20	2.20E+01 \pm 2.17E-01	2.13E+01 \pm 2.40E-01	20	5.00E+01 \pm 2.67E-12	5.00E+01 \pm 8.89E-13
21	3.22E+02 \pm 9.23E+01	3.07E+02 \pm 7.92E+01	21	3.83E+02 \pm 3.41E+02	6.42E+02 \pm 4.45E+02	21	6.67E+02 \pm 6.05E+01	3.24E+02 \pm 4.28E+01
22	6.40E+03 \pm 4.89E+02	6.56E+03 \pm 3.44E+02	22	1.14E+04 \pm 1.09E+03	1.25E+04 \pm 6.39E+02	22	3.02E+04 \pm 5.49E+02	3.02E+04 \pm 4.48E+02
23	7.05E+03 \pm 2.71E+02	6.94E+03 \pm 3.42E+02	23	1.36E+04 \pm 3.69E+02	1.34E+04 \pm 4.34E+02	23	3.07E+04 \pm 4.37E+02	3.03E+04 \pm 4.73E+02
24	2.00E+02 \pm 8.79E-01	2.01E+02 \pm 1.17E+00	24	2.05E+02 \pm 8.25E+00	2.21E+02 \pm 8.96E+00	24	3.40E+02 \pm 1.07E+01	2.02E+02 \pm 1.76E+00
25	2.39E+02 \pm 4.07E+00	2.43E+02 \pm 4.78E+00	25	2.75E+02 \pm 7.63E+00	2.84E+02 \pm 6.79E+00	25	6.28E+02 \pm 3.58E+01	3.57E+02 \pm 1.10E+01
26	2.02E+02 \pm 1.40E+01	2.02E+02 \pm 1.42E+01	26	2.66E+02 \pm 5.85E+01	2.52E+02 \pm 6.27E+01	26	6.88E+02 \pm 8.56E+00	3.23E+02 \pm 1.15E+01
27	3.12E+02 \pm 3.74E+01	3.12E+02 \pm 3.25E+01	27	5.12E+02 \pm 1.46E+02	5.94E+02 \pm 1.18E+02	27	2.92E+03 \pm 3.71E+02	3.56E+02 \pm 4.38E+01
28	3.00E+02 \pm 2.91E-13	3.00E+02 \pm 3.20E-13	28	4.58E+02 \pm 4.16E+02	4.00E+02 \pm 2.87E-13	28	3.96E+03 \pm 1.15E+03	3.52E+03 \pm 1.06E+03
Significantly better/equal/worse		12+/11=-/5-	Significantly better/equal/worse		15+/5=-/8-	Significantly better/equal/worse		22+/6=-/0-

where u'_{ij} is the j th dimension of the i th new individual generated by (4), and $ubound$ and $lbound$ are the upper and lower bounds, respectively.

When developing a DEEP algorithm from JADE, F , and sCR in (4) remain associated with each individual, as used in the original DE. Two minor modifications are however, made in JADEEP for the DEEP framework. First, when we use the sCR scale factor to control the EP information in (4), a relatively large sCR can be used of when CR is large, for example

$$sCR_i = \begin{cases} 1, & \text{if } CR_i > 0.2 \\ CR_i, & \text{otherwise} \end{cases} \quad (14)$$

because a large CR usually indicates that the algorithm is progressing well and the added EP information should be used more. The second modification involves the anchor, c_{EP} , which is used as a promising point to guide the JADE mutation. It is hence better to decrease the p -best guidance in the original JADE mutation as follows:

$$v_i = x_i + F'_i \cdot (x_{pbest} - x_i) + F_i \cdot (x_{r_1} - x_{r_2}) \quad (15)$$

where

$$F'_i = F_i \cdot (1 - \beta_i \cdot sCR_i). \quad (16)$$

The scale factor F' is used to control how the p -best guidance information is decreased in relation to the mount added by (4), so as to balance JADEEP for various kinds of optimization problems.

A full pseudo code of JADEEP is shown in the Appendix. Details of the parameters used in these tests, as well as in the original DE/rand/1 algorithm, are given in Table I. The original settings of JADE remain unchanged. The settings of the control parameters used in the self-adaptation of the EP mutation are also shown in Table I.

We endeavor to use the same settings for all the test algorithms. The initial values of α_m and β_m are set to zero, which means no EP guidance is used at the very beginning. Empirically, α_{sig} and β_{sig} are set to a smaller value than the search range, for tests advise that they work fine as long as no extreme values are used. We shall conduct experiments to test how the range control parameters α_{max} and β_{max} influence DEEP performance and self-adaptation of α_i and β_i . We also conduct experiments on the anchor point learning parameter λ and the population center calculation parameter s .

B. Comparing the DEEP Algorithms With Their Original DEs

DE/rand/1/EP and JADEEP are compared with their original DE algorithms in Tables II and III, respectively, on 30, 50, and 100 dimensional test problems of CEC'13. All test results are based on 51 independent runs. The results with better mean fitness errors compared with the corresponding results of the original algorithms are shown in boldface, while the results which are significantly different (determined by the Wilcoxon

TABLE III

TEST RESULTS OF JADEEP VERSUS JADE IN TEST PROBLEMS WITH $D = 30, 50$, AND 100 (NUMBER OF FES IS SHOWN IN PARENTHESES IF MEAN ERROR IS ZERO; BETTER MEAN ERRORS ARE SHOWN IN BOLDFACE; SIGNIFICANTLY DIFFERENT RESULTS ARE HIGHLIGHTED IN SHADE)

D	JADE	JADEEP	D	JADE	JADEEP	D	JADE	JADEEP
30	Mean \pm std	Mean \pm std	50	Mean \pm std	Mean \pm std	100	Mean \pm std	Mean \pm std
1	0 (3.37E+04 \pm 9.68E+02)	0 (2.66E+04 \pm 3.84E+02)	1	0 (4.08E+04 \pm 1.03E+03)	0 (3.15E+04 \pm 5.46E+02)	1	0 (1.82E+05 \pm 2.83E+03)	0 (1.36E+05 \pm 1.23E+03)
2	7.79E+03 \pm 5.61E+03	5.43E+03 \pm 4.22E+03	2	2.53E+04 \pm 1.52E+04	1.19E+04 \pm 5.35E+03	2	1.61E+05 \pm 4.07E+04	8.28E+04 \pm 2.18E+04
3	2.97E+05 \pm 1.68E+06	6.19E+06 \pm 7.51E+06	3	1.82E+06 \pm 4.02E+06	4.25E+07 \pm 4.73E+07	3	1.05E+07 \pm 8.79E+06	6.63E+07 \pm 4.55E+07
4	6.98E+03 \pm 1.47E+04	5.00E+03 \pm 1.03E+04	4	5.55E+03 \pm 1.71E+04	5.12E+03 \pm 1.42E+04	4	3.70E+04 \pm 7.58E+04	4.23E+04 \pm 4.77E+04
5	0 (4.94E+04 \pm 1.49E+03)	0 (3.67E+04 \pm 5.23E+02)	5	0 (6.72E+04 \pm 1.72E+03)	0 (4.56E+04 \pm 9.57E+02)	5	0 (3.39E+05 \pm 6.27E+03)	0 (2.01E+05 \pm 1.28E+03)
6	1.11E+00 \pm 5.19E+00	9.09E-01 \pm 3.65E+00	6	4.36E+01 \pm 7.95E-01	4.36E+01 \pm 7.95E-01	6	1.43E+02 \pm 5.96E+01	9.80E+01 \pm 7.51E+01
7	2.78E+00 \pm 3.22E+00	4.67E+00 \pm 3.72E+00	7	2.05E+01 \pm 9.85E+00	2.10E+01 \pm 7.83E+00	7	1.41E+01 \pm 3.85E+00	1.93E+01 \pm 5.70E+00
8	2.09E+01 \pm 8.42E-02	2.09E+01 \pm 4.90E-02	8	2.11E+01 \pm 1.05E-01	2.11E+01 \pm 3.30E-02	8	2.13E+01 \pm 2.56E-02	2.13E+01 \pm 3.52E-02
9	2.69E+01 \pm 1.26E+00	2.69E+01 \pm 1.63E+00	9	5.41E+01 \pm 2.47E+00	5.42E+01 \pm 2.07E+00	9	1.33E+02 \pm 2.86E+00	1.33E+02 \pm 3.04E+00
10	3.96E-02 \pm 2.42E-02	3.75E-02 \pm 1.82E-02	10	3.21E-02 \pm 2.06E-02	3.76E-02 \pm 2.89E-02	10	1.89E-02 \pm 1.69E-02	2.07E-02 \pm 1.56E-02
11	0 (1.41E+05 \pm 1.96E+03)	0 (9.90E+04 \pm 1.60E+03)	11	0 (2.31E+05 \pm 3.66E+03)	0 (1.57E+05 \pm 3.28E+03)	11	8.55E+00 \pm 9.15E-01	5.63E+00 \pm 7.46E-01
12	2.49E+01 \pm 4.43E+00	2.06E+01 \pm 3.95E+00	12	5.81E+01 \pm 1.18E+01	5.18E+01 \pm 6.56E+00	12	2.67E+02 \pm 1.95E+01	2.65E+02 \pm 1.55E+01
13	4.67E+01 \pm 1.11E+01	4.16E+01 \pm 1.36E+01	13	1.26E+02 \pm 1.87E+01	1.19E+02 \pm 2.60E+01	13	4.15E+02 \pm 3.78E+01	4.28E+02 \pm 2.91E+01
14	2.33E-02 \pm 2.11E-02	4.39E-02 \pm 3.18E-02	14	3.75E-02 \pm 2.12E-02	1.65E-01 \pm 2.89E-02	14	2.14E+03 \pm 1.44E+02	2.63E+03 \pm 1.84E+02
15	3.30E+03 \pm 3.32E+02	3.20E+03 \pm 3.31E+02	15	7.07E+03 \pm 4.89E+02	6.89E+03 \pm 4.82E+02	15	2.30E+04 \pm 4.96E+02	2.23E+04 \pm 6.38E+02
16	1.73E+00 \pm 6.77E-01	1.75E+00 \pm 6.59E-01	16	2.15E+00 \pm 7.89E-01	1.85E+00 \pm 6.61E-01	16	3.19E+00 \pm 6.43E-01	3.23E+00 \pm 6.29E-01
17	3.04E+01 \pm 1.14E-14	3.04E+01 \pm 1.54E-11	17	5.08E+01 \pm 5.40E-14	5.08E+01 \pm 5.50E-14	17	1.27E+02 \pm 1.04E+00	1.32E+02 \pm 1.51E+00
18	7.78E+01 \pm 6.68E+00	7.31E+01 \pm 6.06E+00	18	1.44E+02 \pm 1.10E+01	1.34E+02 \pm 9.02E+00	18	5.66E+02 \pm 1.68E+01	5.56E+02 \pm 1.75E+01
19	1.44E+00 \pm 1.04E-01	1.43E+00 \pm 1.19E-01	19	2.76E+00 \pm 2.16E-01	2.72E+00 \pm 2.30E-01	19	1.35E+01 \pm 6.78E-01	1.32E+01 \pm 6.05E-01
20	1.04E+01 \pm 5.35E-01	1.01E+01 \pm 6.04E-01	20	1.97E+01 \pm 6.04E-01	1.95E+01 \pm 6.00E-01	20	5.00E+01 \pm 3.88E-12	5.00E+01 \pm 1.27E-11
21	2.82E+02 \pm 7.98E+01	2.98E+02 \pm 5.52E+01	21	8.75E+02 \pm 3.40E+02	8.52E+02 \pm 3.49E+02	21	3.78E+02 \pm 4.15E+01	3.88E+02 \pm 3.25E+01
22	8.78E+01 \pm 3.45E+01	1.03E+02 \pm 2.37E+01	22	1.55E+01 \pm 2.07E+01	2.18E+01 \pm 3.75E+01	22	2.99E+03 \pm 5.74E+02	3.12E+03 \pm 6.36E+02
23	3.43E+03 \pm 4.31E+02	3.25E+03 \pm 4.11E+02	23	7.28E+03 \pm 6.71E+02	7.03E+03 \pm 7.83E+02	23	2.35E+04 \pm 7.95E+02	2.23E+04 \pm 7.42E+02
24	2.13E+02 \pm 1.11E+01	2.10E+02 \pm 6.66E+00	24	2.52E+02 \pm 2.47E+01	2.48E+02 \pm 1.35E+01	24	2.53E+02 \pm 1.00E+01	2.56E+02 \pm 7.92E+00
25	2.73E+02 \pm 1.24E+01	2.63E+02 \pm 1.57E+01	25	3.54E+02 \pm 2.23E+01	3.19E+02 \pm 3.01E+01	25	5.98E+02 \pm 1.14E+01	4.67E+02 \pm 7.60E+01
26	2.22E+02 \pm 5.10E+01	2.09E+02 \pm 2.96E+01	26	3.52E+02 \pm 1.01E+02	3.06E+02 \pm 8.68E+01	26	4.87E+02 \pm 1.13E+02	3.68E+02 \pm 9.45E+00
27	6.98E+02 \pm 2.23E+02	5.34E+02 \pm 2.24E+02	27	1.37E+03 \pm 3.64E+02	1.07E+03 \pm 3.49E+02	27	9.61E+02 \pm 1.35E+02	1.05E+03 \pm 1.66E+02
28	3.20E+02 \pm 1.44E+02	3.00E+02 \pm 2.67E-13	28	4.58E+02 \pm 4.17E+02	4.59E+02 \pm 4.21E+02	28	3.20E+03 \pm 9.97E+02	2.96E+03 \pm 8.55E+02
Significantly better/equal/worse		15+/8-/5-	Significantly better/equal/worse		12+/13-/3-	Significantly better/equal/worse		11+/13-/4-

rank-sum test with a significance level 0.05) are highlighted in shade. In the last row of each table, the number of functions on which the DEEP algorithm performs significantly better than, similar to, and significantly worse than the original DE algorithm are presented.

For each test function, the mean absolute errors between the final best fitness and the theoretical best fitness and the standard deviation over the given number of test runs are presented in the tables. The termination condition is the maximum number of function evaluations (FEs), set to be 10^4 per dimension, i.e., for a maximum total of $D \times (1E+4)$ FEs. If the algorithm converge to an absolute error smaller than $1E-9$ (taken as zero) on the CEC13' benchmarks, the algorithm terminates and the total number of FEs used are recorded and shown in parentheses.

Compared with DE/rand/1, the DEEP algorithm has been able to improve the performance on most of F1–F20 functions, while maintaining comparable performance on the composite functions F21–F28, as shown in Table II. The improvements become more salient when the dimensionality of the problems increases. The DEEP algorithm has worked especially well for most functions when D increases to 100 (with a population size 400). One may be interested to measure how much the DEEP improves on such a simple function as F1, when performance on multimodal functions are at least not worse. It is seen that the numbers of FEs were reduced to 46% and 41% for $D = 30$ and $D = 50$, respectively. For $D = 100$, the DE/rand/EP managed to find the theoretical minimum of

F1 with 56% of FEs of the original algorithm, which failed to reach a zero error and was terminated at the end of the predefined maximum FEs.

Similarly, JADEEP is also able to improve the performance over JADE, not only on unimodal functions, but also on composite functions, as shown in Table III. The JADEEP runs much faster than JADE on simple functions like F5, with FEs reduced to 60% when $D = 100$. Note that the acceleration is gained while JADEEP can perform well on complicated multimodal functions like F23 and F26. This is owing to EP providing good but not greedy guidance. For a more direct view on the improvements brought about by the DEEP framework, a search performance chart comparing JADE and JADEEP on some typical functions ($D = 30$) is shown in Fig. 5.

We also tested the application of EP to ensemble-based DE variants with multiple strategies, such as SaDE, CoDE, and SaEPSDE, although EP showed less significant improvements. The reason might be that the greedy mutation strategy DE/best used in SaDE weakens the guidance effects of DEEP and the competing mechanism of mutation strategies in these algorithms weakens the effects of EP. We would hence propose in future work that dedicated EP-based mutation strategies be designed to work with the ensemble-based DE algorithms.

C. Functional Analysis on DEEP With Parameter Studies

In this sub-section, we investigate how EP functions with key parameters in the DEEP framework. By testing different

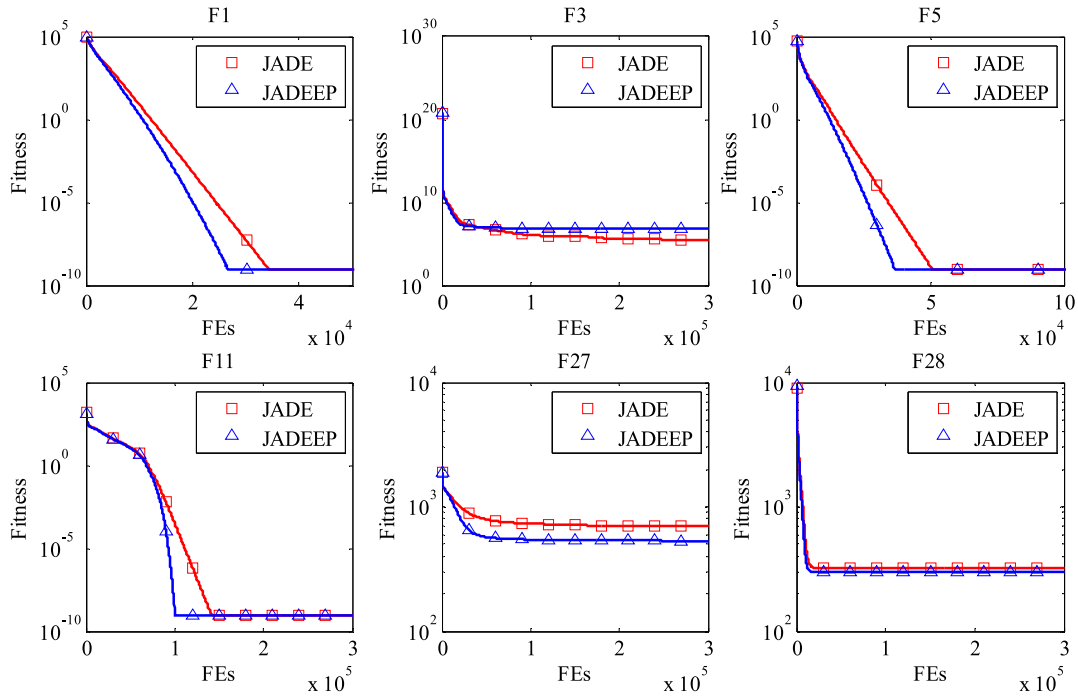


Fig. 5. Comparison of evolution processes between JADEEP and JADE on some representative functions.

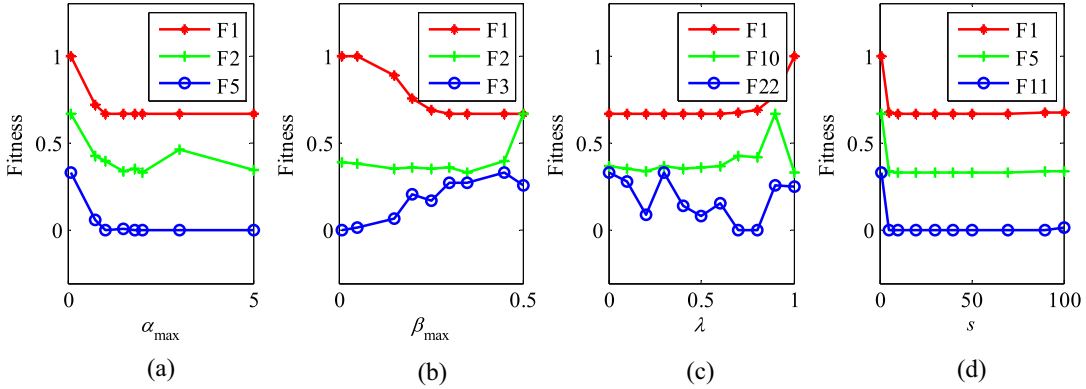


Fig. 6. Parameters study on (a) α_{\max} , (b) β_{\max} , (c) λ , and (d) s .

values of certain key parameters, the role of the two parts of EP can be more clearly understood.

The EP vector v_{EP} is designed to help the population in long migration, and α_{\max} is used to control possible maximum effects of v_{EP} . The effects of v_{EP} can be much clearer if the initial populations happen to be far away from the global best region, especially when a promising range of the global best is unknown. We study such cases with different settings of α_{\max} and with the population initialized far from the global best. The search results on some functions are shown in Fig. 6(a) (scaled for the convenience of comparison). The results show that with a larger α_{\max} , the search speed can be improved noticeably. This confirms that the EP vector v_{EP} can be helpful especially when a long migration is necessary. Thus we have recommended a large α_{\max} as a suitable setting in Table I.

The EP anchor point c_{EP} is designed to help the population reduce wasteful detours in the search progress. In (4) the

mutation vector is affected by $c_{EP} - u_i$, being dragged closer to c_{EP} . Too much influence of c_{EP} would cause the population to converge prematurely, which is undesirable even when c_{EP} is different from the current best location. Thus parameter β_{\max} , which is used to control the effects of c_{EP} , must be set to a proper value. As shown in Fig. 6(b), different settings of β_{\max} are tested on several functions. To make a balance on search speed and divergence, we choose a median value of 0.25 for β_{\max} as used in Table I. For the anchor learning parameter λ , in Fig. 6(c), test results guided us to select a value of 0.5, as used in Table I. Setting λ larger than 0 means that more information than the current population center is used, which leads to better performance on some functions. On the other hand, setting λ to a value closer to 1 leads to poor results in most cases, because the anchor point can be far from the global best and may not change during the evolution process. However, it can be good for multimodal functions, as shown in Fig. 6(c) on F10, because the anchor can be helpful

TABLE IV
JADEEP COMPARED WITH DCMA-EA, CMA-ES, AND DE/BEST/2 (NUMBER OF FES IS SHOWN IN PARENTHESES IF MEAN ERROR IS ZERO;
BETTER MEAN ERRORS ARE SHOWN IN BOLDFACE; SIGNIFICANTLY DIFFERENT RESULTS ARE HIGHLIGHTED IN SHADE)

D	JADEEP	DCMA-EA	CMA-ES	DE/best/2
30	Mean±std	Mean±std	Mean±std	Mean±std
1	0 (2.66E+04±3.84E+02)	0 (3.30E+04±7.05E+02)	0 (7.90E+03±4.79E+02)	2.42E+02±1.56E+02
2	5.43E+03±4.22E+03	6.08E+04±3.49E+04	0 (6.41E+04±8.31E+02)	6.58E+06±3.88E+06
3	6.19E+06±7.51E+06	1.27E+03±6.42E+03	6.81E+02±3.32E+03	8.81E+08±6.68E+08
4	5.00E+03±1.03E+04	1.91E+01±2.54E+01	0 (8.76E+04±1.04E+03)	1.77E+04±9.38E+03
5	0 (3.67E+04±5.23E+02)	0 (6.04E+04±2.87E+03)	0 (1.55E+05±1.12E+04)	3.22E+01±2.75E+01
6	9.09E-01±3.65E+00	1.63E+00±6.28E+00	3.11E+00±8.59E+00	5.78E+01±2.56E+01
7	4.67E+00±3.72E+00	4.95E+00±4.53E+00	1.50E+01±1.01E+01	3.78E+01±1.62E+01
8	2.09E+01±4.90E-02	2.09E+01±4.59E-02	2.15E+01±8.88E-02	2.08E+01±1.33E-01
9	2.69E+01±1.63E+00	2.00E+01±7.29E+00	4.14E+01±8.36E+00	2.33E+01±4.31E+00
10	3.75E-02±1.82E-02	2.11E-02±1.29E-02	1.70E-02±1.11E-02	6.31E+01±4.31E+01
11	0 (9.90E+04±1.60E+03)	1.67E+02±1.07E+01	5.81E+01±8.56E+01	6.18E+01±1.36E+01
12	2.06E+01±3.95E+00	1.88E+02±1.23E+01	8.08E+02±1.07E+03	1.50E+02±2.39E+01
13	4.16E+01±1.36E+01	1.99E+02±9.01E+01	1.76E+03±1.60E+03	1.75E+02±2.86E+01
14	4.39E-02±3.18E-02	3.89E+03±8.55E+02	5.15E+03±9.96E+02	1.56E+03±3.42E+02
15	3.20E+03±3.31E+02	4.59E+03±7.75E+02	5.15E+03±8.19E+02	5.57E+03±6.35E+02
16	1.75E+00±6.59E-01	1.90E+00±8.48E-01	9.43E-02±7.23E-02	1.31E+00±7.34E-01
17	3.04E+01±1.54E-11	1.98E+02±1.24E+01	3.94E+03±6.85E+02	1.55E+02±3.98E+01
18	7.31E+01±6.06E+00	2.19E+02±1.17E+01	4.16E+03±9.08E+02	2.48E+02±2.94E+01
19	1.43E+00±1.19E-01	1.57E+01±9.86E-01	3.27E+00±6.56E-01	2.24E+01±3.57E+01
20	1.01E+01±6.04E-01	1.49E+01±4.45E-01	1.25E+01±8.69E-01	1.23E+01±4.39E-01
21	2.98E+02±5.52E+01	3.07E+02±8.29E+01	3.12E+02±8.36E+01	5.53E+02±1.19E+02
22	1.03E+02±2.37E+01	5.71E+03±1.26E+03	6.94E+03±8.33E+02	1.56E+03±4.14E+02
23	3.25E+03±4.11E+02	6.11E+03±9.05E+02	6.47E+03±8.20E+02	5.84E+03±7.63E+02
24	2.10E+02±6.66E+00	2.31E+02±3.63E+01	7.94E+02±6.77E+02	2.53E+02±1.11E+01
25	2.63E+02±1.57E+01	2.51E+02±1.55E+01	3.18E+02±1.60E+02	2.62E+02±7.41E+00
26	2.09E+02±2.96E+01	2.62E+02±1.01E+02	4.56E+02±4.51E+02	2.56E+02±7.27E+01
27	5.34E+02±2.24E+02	5.90E+02±1.37E+02	5.67E+02±1.83E+02	8.49E+02±1.10E+02
28	3.00E+02±2.67E-13	4.02E+02±3.12E+02	1.78E+03±3.44E+03	8.73E+02±2.10E+02
No. of JADEEP is better/equal/worse		17+/6-/5-	20+/2-/6-	26+/2-/0-

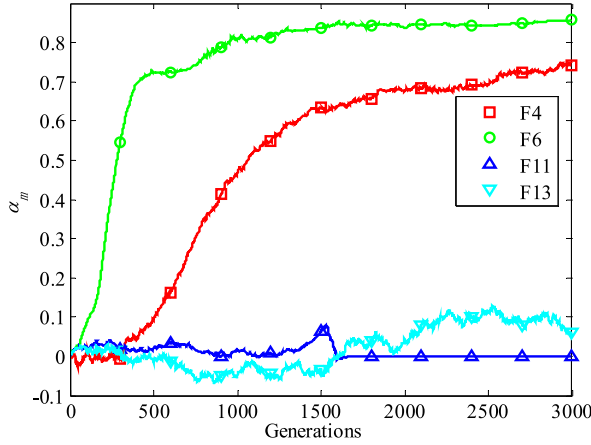


Fig. 7. Self-adaptation of α_m in the search process.

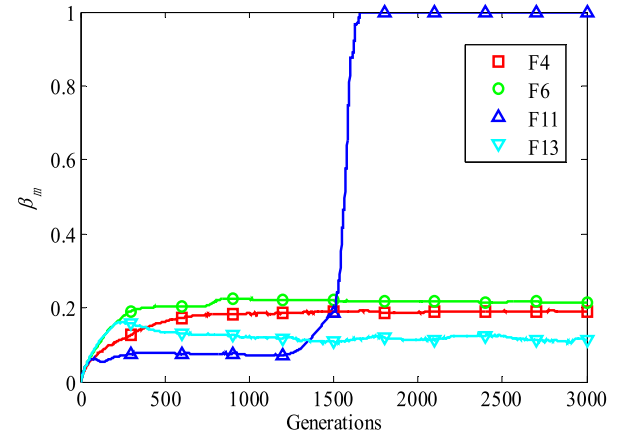


Fig. 8. Self-adaptation of β_m in the search process.

in creating divergence when it is far from the current local best. This explains why the DEEP is able to help improve the performance on the multimodal functions, with a possibility that the c_{EP} guidance can be quite different from the current local best.

Another key feature of the DEEP algorithms is the population center. Parameter s is used to compute the center of the population (center of the first s best individuals) and is studied in Fig. 6(d). Using either the current best individual ($s = 1$) or the whole population ($s = 100$ for JADE) is not as good as using our recommended value $s = 20$. The test results show

that the population center can make a better effect of guidance than the current best in DEEP. It is also reasonable to use not all the individuals in the population to calculate the population center because some of the individuals in the population are only used to maintain divergence.

The self-adaptation mechanism is profoundly used in the literature. For other parameters in the self-adaptation, Table I also lists their empirical values. The initial values of α_m and β_m are set to 0, which means at the beginning the EP is not used. Figs. 7 and 8 show the self-adaptation curve of α_m

TABLE V
JADEEP COMPARED WITH SADE, CoDE, SaEPSDE, AND SANSDE (NUMBER OF FES IS SHOWN IN PARENTHESES IF MEAN ERROR IS ZERO;
BETTER MEAN ERRORS ARE SHOWN IN BOLDFACE; SIGNIFICANTLY DIFFERENT RESULTS ARE HIGHLIGHTED IN SHADE)

D	JADEEP	SaDE	CoDE	SaEPSDE	SaNSDE
30	Mean±std	Mean±std	Mean±std	Mean±std	Mean±std
1	0 (2.66E+04±3.84E+02)	0 (2.77E+04±8.51E+02)	0 (5.73E+04±1.33E+03)	0 (3.99E+04±1.05E+03)	0 (7.85E+04±1.56E+03)
2	5.43E+03±4.22E+03	3.94E+04±2.33E+04	7.15E+04±3.71E+04	3.48E+04±2.20E+04	4.25E+04±3.23E+04
3	6.19E+06±7.51E+06	2.32E+06±4.42E+06	4.99E+05±1.41E+06	7.46E+06±1.60E+07	1.24E+05±7.26E+05
4	5.00E+03±1.03E+04	3.72E+02±4.59E+02	7.02E-02±1.07E-01	2.89E+01±3.30E+01	1.03E+03±6.12E+03
5	0 (3.67E+04±5.23E+02)	0 (4.58E+04±1.78E+03)	0 (7.59E+04±1.61E+03)	0 (5.26E+04±1.46E+03)	0 (1.04E+05±2.33E+03)
6	9.09E-01±3.65E+00	6.49E+00±5.61E+00	2.62E+00±7.03E+00	5.79E+00±1.03E+01	3.94E+00±7.54E+00
7	4.67E+00±3.72E+00	1.75E+01±1.14E+01	9.97E+00±7.24E+00	4.59E+01±1.24E+01	8.68E+00±1.77E+01
8	2.09E+01±4.90E-02	2.09E+01±5.54E-02	2.08E+01±1.10E-01	2.09E+01±7.53E-02	2.09E+01±4.94E-02
9	2.69E+01±1.63E+00	1.66E+01±2.89E+00	1.43E+01±3.51E+00	2.38E+01±4.51E+00	2.77E+01±1.56E+00
10	3.75E-02±1.82E-02	1.37E-01±7.11E-02	3.91E-02±2.18E-02	1.11E-01±8.23E-02	3.42E-02±2.06E-02
11	0 (9.90E+04±1.60E+03)	7.80E-02±2.70E-01	1.95E-02±1.39E-01	3.90E-02±2.79E-01	0 (1.78E+05±5.31E+03)
12	2.06E+01±3.95E+00	4.22E+01±1.07E+01	3.82E+01±1.12E+01	6.50E+01±1.93E+01	5.38E+01±1.05E+01
13	4.16E+01±1.36E+01	8.64E+01±1.94E+01	8.12E+01±2.81E+01	1.24E+02±2.59E+01	8.00E+01±1.53E+01
14	4.39E-02±3.18E-02	7.93E-01±1.11E+00	3.16E+00±2.94E+00	6.00E+00±2.35E+01	1.41E+00±5.60E-01
15	3.20E+03±3.31E+02	4.73E+03±9.16E+02	3.25E+03±4.84E+02	3.90E+03±5.84E+02	4.57E+03±4.03E+02
16	1.75E+00±6.59E-01	2.25E+00±2.98E-01	3.62E-01±2.58E-01	1.38E+00±3.91E-01	1.94E+00±3.49E-01
17	3.04E+01±1.54E-11	3.05E+01±7.87E-02	3.04E+01±1.24E-02	3.04E+01±6.91E-03	3.06E+01±6.19E-02
18	7.31E+01±6.06E+00	1.27E+02±2.87E+01	6.73E+01±1.14E+01	7.62E+01±1.70E+01	1.45E+02±1.45E+01
19	1.43E+00±1.19E-01	3.47E+00±5.56E-01	1.59E+00±3.99E-01	3.86E-01±2.02E-01	2.11E+00±1.55E-01
20	1.01E+01±6.04E-01	1.07E+01±6.35E-01	1.05E+01±7.88E-01	1.15E+01±6.54E-01	1.15E+01±3.83E-01
21	2.98E+02±5.52E+01	3.32E+02±7.82E+01	2.93E+02±9.86E+01	2.33E+02±4.76E+01	2.78E+02±7.72E+01
22	1.03E+02±2.37E+01	1.15E+02±8.87E+00	1.13E+02±1.98E+01	6.93E+01±4.86E+01	1.77E+02±3.26E+01
23	3.25E+03±4.11E+02	4.82E+03±9.54E+02	3.58E+03±6.49E+02	4.57E+03±8.20E+02	4.78E+03±4.97E+02
24	2.10E+02±6.66E+00	2.20E+02±6.36E+00	2.20E+02±6.77E+00	2.64E+02±1.05E+01	2.20E+02±2.19E+01
25	2.63E+02±1.57E+01	2.61E+02±1.39E+01	2.55E+02±7.75E+00	2.77E+02±1.06E+01	2.84E+02±1.08E+01
26	2.09E+02±2.96E+01	2.05E+02±2.47E+01	2.02E+02±1.65E+01	2.00E+02±1.65E-03	2.00E+02±2.06E-03
27	5.34E+02±2.24E+02	5.54E+02±6.61E+01	5.76E+02±1.05E+02	7.81E+02±2.53E+02	9.55E+02±7.78E+01
28	3.00E+02±2.67E-13	3.00E+02±0.00E+00	3.00E+02±0.00E+00	2.92E+02±3.92E+01	3.00E+02±0.00E+00
No. of JADEEP is better/equal/worse		15+/11=-/2-	10+/13=-/5-	16+/8=-/4-	17+/9=-/2-

and β_m during the evolution process of JADEEP on functions F4, F6, F11, and F13. These curves show these two control parameters behave differently for different problems. Based on Fig. 7, the self-adaptation mechanism seems to be useful to find different values of α_m , large or small, negative or positive, which are proper for different functions. For example, for function F6 (Rosenbrock) with a long descent corridor that would require many generations of evolution, a large α_m is found to be necessary. Fig. 8 also shows that different settings of β_m emerge for different functions during the evolution process.

D. Comparing JADEEP With Other Relevant Algorithms

In this sub-section, the DCMA-EA [21] algorithm that directly hybridizes the DE and CMA-ES is compared in detail. More competitors include CMA-ES itself, which is a state-of-the-art CM algorithm, the DE/best/2, which can also be taken as a CM algorithm because all individuals are generated based on the current best individual, and a number of state-of-the-art DE variants such as SaDE, CoDE, SaEPSDE, and SaNSDE, which use ensemble mutation strategies. Nonparametric Dunnett's test [53] for multicomparisons is used here with JADEEP taken as the control algorithm against all other algorithms shown in Tables IV and V. The results that are significantly different (with a significance level 0.05) are highlighted in shade as before.

TABLE VI
COMPARISON ON TIME OVERHEADS BETWEEN JADEEP, DCMA-EA, AND CMA-ES

Test functions	Times used (Seconds)		
	JADEEP	DCMA-EA	CMA-ES
F1(3E+5 FEs)	106.30	196.78	225.40
F2(3E+5 FEs)	191.04	284.00	298.28
F3(3E+5 FEs)	241.35	345.83	369.51
F4(3E+5 FEs)	161.77	255.88	277.23

From the test results shown in Table IV, we can see that the proposed JADEEP performs better than DCMA-EA and CMA-ES on most of the multimodal functions. The search speed of JADEEP is also higher than that of DCMA-EA. In DCMA-EA, EP information is used after an eigen-decomposition, which can be time consuming, as shown in Table VI comparing the computational overheads of JADEEP, DCMA-EA, and CMA-ES. These are based on 51 independent runs of each function on a computer with Intel Core i3-3240 CPU at 3.4 GHz and 4 GB memory and the MATLAB 2013a environment. We can see that JADEEP runs much faster, because the EP computation is in the order of $O(1)$ after the computation of the population center while the eigen-decomposition used in DCMA-EA and CMA-ES is $O(n^3)$ for a n -dimensional problem.

TABLE VII
TEST RESULTS ON LENNARD-JONES AND TERSOFF POTENTIAL
OPTIMIZATION PROBLEMS

Algorithms	Lennard-Jones	Tersoff
	Mean±std	Mean±std
CoDE	-26.6404±2.027502	-20.0118±2.892568
SaDE	-22.2263±2.406471	-18.1598±2.844712
SaEPSDE	-25.6057±3.422694	-23.7447±5.939077
SaNSDE	-20.2214±1.305379	-19.2292±4.552788
CMAES	-17.2145±4.587383	-18.1506±4.387988
DE/best	-11.6457±3.141150	-19.1920±3.365700
DCMAEA	-6.66565±0.958844	-19.2574±5.241443
JADE	-22.8183±0.853603	-29.0736±0.069556
JADEEP	-26.7473±0.755451	-29.1622±0.003269

```

Pseudo code of JADEEP
Begin
  Set mCR=0.5; mF=0.5. A=empty set.
  Set  $\alpha_m=0$ ;  $\beta_m=0$ ;
  Create a random initial population  $P = \{x_{i,0} | i=1, \dots, NP\}$ 
  Set  $v_{EP}$ =zero vector,  $c_{EP}$ = center of the population;
  For g=1 to G
    SF=empty set; SCR=empty set;
     $\alpha_{good}$ =empty set;  $\beta_{good}$ =empty set;
    For i=1 to NP
      Generate  $CR_i = \text{randn}(mCR, 0, 1)$ ,  $F_i = \text{randc}(mF, 0, 1)$ 
      Generate  $\alpha_i$  and  $\beta_i$  according to (9) and (11).
      Randomly choose  $x_{pbest,g}$  as one of the 100p% best vectors
      Randomly choose  $x_{r1,g} \sim x_{i,g}$  from P
      Randomly choose  $x_{r2,g} \sim x_{r1,g} \sim x_{i,g}$  from PUA
      Generate  $sCR_i$  according to (14)
      Generate  $F_i$  according to (16)
      %%The following modification is not applied to DE/rand/1
       $v_{i,g} = x_{i,g} + F_i * (x_{pbest,g} - x_{i,g}) + F_i * (x_{r1,g} - x_{r2,g})$ 
      Generate  $jrand = \text{randint}(1, D)$ 
      For j=1:D
        If  $j=jrand$  or  $\text{rand}(0,1) < CR_i$ 
           $u_{j,i,g} = v_{j,i,g}$ 
        Else
           $u_{j,i,g} = x_{j,i,g}$ 
        End if
      End for
      %%EP added here
      %%mutation operations.
      Modify  $u_{i,g}$  according to (4);
      Handle bound constraint;
      If  $f(x_{i,g}) < f(u_{i,g})$ 
         $x_{i,g+1} = u_{i,g}$ 
      Else
         $x_{i,g+1} = u_{i,g}$ ;  $x_{i,g} \rightarrow A$ ;  $CR_i \rightarrow SCR$ ;  $F_i \rightarrow SF$ ;
         $\alpha_i \rightarrow \alpha_{good}$ ;  $\beta_i \rightarrow \beta_{good}$ 
      End if
    End for
    Randomly remove solutions from A so that  $|A| \leq NP$ 
     $mCR = (1-c)*mCR + c*\text{mean}_A(SCR)$ 
     $mF = (1-c)*mF + c*\text{mean}_A(SF)$ 
    Update  $\alpha_m$  and  $\beta_m$  according to (10) (12)
    Update  $v_{EP}$  and  $c_{EP}$  according to (5)(6)(7)
  End for
End

```

Fig. 9. Pseudo code of JADEEP.

The comparison in Table V shows that JADEEP works better than SaDE and SaNSDE on most functions. SaDE is already a fast DE algorithm and is better than the original JADE, which is especially good at unimodal functions. With the help of the EP, JADEEP improves its performance on unimodal functions while maintaining advantages on multimodal functions. JADEEP is also seen slightly better than SaEPSDE and CoDE on multimodal functions and is much faster than the two algorithms on unimodal functions (such as F1, F2, and F5). However, EP is less useful in JADEEP on some unimodal functions such as F3, as the original JADE algorithm itself is not good at this function.

E. Comparison of JADEEP Applied to Practical Problems

Here JADEEP is applied to practical optimization problems studied in [52] which are atomic potential minimization problems, important in molecular structure studies. Two potential function minimization problems were studied: the Lennard-Jones potential problem and the Tersoff potential function minimization problem. For both problems, the goal is to find the positions of a series of N atoms

$$\vec{X} = \{\vec{X}_1, \vec{X}_2, \dots, \vec{X}_N\} \quad (17)$$

such that the atomic potential is minimized, where \vec{X}_i is a 3-D position vector. For detailed formulation of these two problems, refer to [52].

In the experiments, JADEEP and other relevant algorithms were compared for $N = 10$ (i.e., 10 atoms with 30 decision variables). As the population size of JADEEP could be reduced to as low as 30, this size was used in the experiments, while all settings for the other relevant algorithms remained unchanged. The results are shown in Table VII. It can be seen that JADEEP performed well and resulted in the lowest potential in these two practical problems.

V. CONCLUSION

In this paper, the reproduction mechanism of the DE has been enhanced through an evolution path, taking the advantages of both a CM of CMA-ES and a DM of DE. Without loss of generality, a Gaussian distribution is used in the CM as a generator for a new population centered at the mean with its path cumulatively learned during the evolutionary process. This adds a momentum in the evolution to speed up search along the trend. In the meanwhile, the DM inherent in DE uses individuals more sparsely distributed in the population without following a parametric model and hence exploration in the evolution is much retained.

Instead of directly combing the DE and the CMA-ES algorithms mechanistically together to develop a new algorithm, only the key CM feature of the CMA-ES, i.e., the evolution path, is used to develop DEEP efficiently for performance improvements. In particular, the direction vector of EP is utilized for additional mutation information and the cumulative learning weighted mean of recent EP centers is used to guide the new individuals generated by the DE. Further, CM features are self-adaptive, which helps new individuals improve quality in a new environment and simplifies algorithm design. Using the DEEP framework, we have developed and illustrated two DEEP algorithms. They have exhibited mostly better performance than the original DE and other relevant state-of-the-art algorithms.

For future work, stand-alone EP-dedicated mutation will be studied. Comparative investigation and combination of parametric and nonparametric reproduction models of EAs are also worthy of further studies, as the results could be very useful in applying EP to other DM algorithms, such as PSO. A CM algorithm enhanced by DM with a direct parent-offspring relationship could be very interesting, too. As surveyed in [54], some machine learning tools may be useful for a possible merger between the CM and DM.

APPENDIX

Detailed pseudo code of JADE with EP modifications (JADEEP) is shown in Fig. 9. More information please contact with the corresponding author Z. H. Zhan (zhazh@mail.sysu.edu.cn).

REFERENCES

- [1] J. Zhang, H. Chung, and W. L. Lo, "Clustering-based adaptive crossover and mutation probabilities for genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 326–335, Jun. 2007.
- [2] R. Storn and K. V. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [3] Z. H. Zhan, J. Zhang, Y. Li, and Y. H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832–847, Dec. 2011.
- [4] B. Xue, M. J. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.
- [5] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362–1381, Dec. 2009.
- [6] W. Chen *et al.*, "Particle swarm optimization with an aging leader and challengers," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 241–258, Apr. 2013.
- [7] Y. H. Li, Z. H. Zhan, S. Lin, R. Wang, and X. N. Luo, "Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems," *Inf. Sci.*, DOI: 10.1016/j.ins.2014.09.030, 2014.
- [8] Z. H. Zhan *et al.*, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.
- [9] M. Shen *et al.*, "Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 7141–7151, Dec. 2014.
- [10] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston, MA, USA: Kluwer Academic, 2001.
- [11] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, 2001.
- [12] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, 2008.
- [13] Z. H. Zhan *et al.*, "An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 399–412, Jun. 2010.
- [14] S. Tsutsui, "cAS: Ant colony optimization with cunning ants," in *Parallel Problem Solving from Nature—PPSN IX* (Lecture Notes in Computer Science), vol. 4193. Berlin, Germany: Springer, 2006, pp. 162–171.
- [15] C. W. Ahn, J. An, and J.-C. Yoo, "Estimation of particle swarm distribution algorithms: Combining the benefits of PSO and EDAs," *Inf. Sci.*, vol. 192, pp. 109–119, Jun. 2012.
- [16] K. Opara and J. Arabas, "Differential mutation based on population covariance matrix," in *Parallel Problem Solving from Nature, PPSN XI* (Lecture Notes in Computer Science), vol. 6238. Berlin, Germany: Springer, 2010, pp. 114–123.
- [17] J. Sun, Q. Zhang, and E. P. K. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," *Inf. Sci.*, vol. 169, nos. 3–4, pp. 249–262, 2005.
- [18] S. Tsutsui, "Probabilistic model-building genetic algorithms in permutation representation domain using edge histogram," in *Parallel Problem Solving from Nature—PPSN VII*. Berlin, Germany: Springer, 2002, pp. 224–233.
- [19] A. Zhou, Q. Zhang, Y. Jin, and B. Sendhoff, "Combination of EDA and DE for continuous biobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 1447–1454.
- [20] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimisation: A novel method for constrained mechanical design optimisation problems," *Comput.-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.
- [21] S. Ghosh, S. Das, S. Roy, S. K. Islama, and P. N. Suganthan, "A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization," *Inf. Sci.*, vol. 182, no. 1, pp. 199–219, 2012.
- [22] K. Deb, A. Anand, and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization," *Evol. Comput.*, vol. 10, no. 4, pp. 345–369, 2002.
- [23] H. Someya, "Striking a mean- and parent-centric balance in real-valued crossover operators," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 737–754, Dec. 2013.
- [24] J. Q. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Jun. 2009.
- [25] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [26] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 1110–1116.
- [27] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.
- [28] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adaptive control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.
- [29] W. J. Yu *et al.*, "Differential evolution with two-level parameter adaptation," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1080–1099, Jul. 2014.
- [30] Y. Wang, Z. X. Cai, and Q.-F. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.
- [31] W. Y. Gong, Z. H. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 397–413, Apr. 2011.
- [32] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "An improved self-adaptive differential evolution algorithm for optimization problems," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 89–99, Feb. 2013.
- [33] R. Mallipeddi and P. N. Suganthan, "Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies," in *Swarm, Evolutionary, and Memetic Computing*. Berlin, Germany: Springer, 2010, pp. 71–78.
- [34] H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian bare-bones differential evolution," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 634–647, Apr. 2013.
- [35] S. Das, A. Abraham, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–552, Jun. 2009.
- [36] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 482–500, Apr. 2012.
- [37] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2066–2081, Dec. 2013.
- [38] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 99–119, Feb. 2011.
- [39] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.
- [40] S. Das, A. Konar, U. K. Chakraborty, and A. Abraham, "Differential evolution with a neighborhood based mutation operator: A comparative study," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.
- [41] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2202–2215, Dec. 2013.
- [42] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, "On convergence of differential evolution over a class of continuous functions with unique global optimum," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 107–124, Feb. 2012.
- [43] Y. Wang and Z. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 117–134, Feb. 2012.
- [44] C. Li and S. Yang, "A general framework of multipopulation methods with clustering in undetectable dynamic environments," *IEEE Trans. Evol. Comput.*, vol. 16, no. 4, pp. 556–577, Aug. 2012.

- [45] S. Das, A. Mandal, and R. Mukherjee, "An adaptive differential evolution algorithm for global optimization in dynamic environments," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 966–978, Jun. 2014.
- [46] F. Peng, K. Tang, G. Chen, and X. Yao, "Multi-start JADE with knowledge transfer for numerical optimization," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 1889–1895.
- [47] J. Sun, J. M. Garibaldi, N. Krasnogor, and Q. Zhang, "An intelligent multi-restart memetic algorithm for box constrained global optimization," *Evol. Comput.*, vol. 21, no. 1, pp. 107–147, 2013.
- [48] A. Basak, S. Das, and K. C. Tan, "Multimodal optimization using a bi-objective differential evolution algorithm enhanced with mean distance based selection," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 666–685, Sep. 2013.
- [49] J. Derrac, S. Garcia, S. Hui, F. Herrera, and P. N. Suganthan, "Statistical analysis of convergence performance throughout the evolutionary search: A case study with SaDE-MMTS and Sa-EPsDE-MMTS," in *Proc. IEEE Symp. Different. Evol.*, Singapore, 2013, pp. 151–156.
- [50] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz. (2014, Jan. 1). "Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization," Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou China, Tech. Rep. 201212 [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm
- [51] S. Das, A. Abraham, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–552, Jun. 2009.
- [52] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems," Dept. Electron. Telecommun. Engg., Jadavpur Univ., Kolkata, India, and School Electr. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep., 2010.
- [53] C. W. Dunnett, "New tables for multiple comparisons with a control," *Biometrics*, vol. 20, no. 1, pp. 482–491, 1964.
- [54] J. Zhang *et al.*, "Evolutionary computation meets machine learning: A survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 68–75, Nov. 2011.



Yuan-Long Li (S'11) received the B.S. degree in mathematics from Sun Yat-sen University, Guangzhou, China, in 2009, where he is currently pursuing the Ph.D. degree.

His current research interests include global optimization, evolutionary algorithm, time series analysis, deep neural network, and their applications in wireless sensor network and data center modeling.



Zhi-Hui Zhan (S'09–M'13) received the bachelor's and Ph.D. degrees from the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently a Lecturer with Sun Yat-sen University. His current research interests include particle swarm optimization, differential evolution, ant colony optimization, genetic algorithms, and their applications in real-world problems.

Dr. Zhan was the recipient of the China Computer Federation Outstanding Dissertation in 2013.



Yue-Jiao Gong (S'10) received the B.S. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2010, where she is currently pursuing the Ph.D. degree.

Her current research interests include artificial intelligence, evolutionary computation, swarm intelligence, and their applications in design and optimization of intelligent transportation systems, wireless sensor networks, and RFID systems.



Wei-Neng Chen (S'07–M'12) received the bachelor's and Ph.D. degrees from the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, in 2006 and 2012, respectively.

He is currently an Associate Professor with the School of Advanced Computing, Sun Yat-sen University. His current research interests include swarm intelligence algorithms and their applications on cloud computing, financial optimization, operations research, and software engineering. He has published over 30 papers in international journals

and conferences.

Prof. Chen was the recipient of the China Computer Federation Outstanding Dissertation in 2012.



Jun Zhang (M'02–SM'08) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Changjiang Chair Professor with the School of Advanced Computing, Sun Yat-sen University, Guangzhou, China. His current research interests include computational intelligence, cloud computing, high performance computing, data mining, wireless sensor networks, operations research, and power electronic circuits. He has published over 100 technical papers in the above areas.

Dr. Zhang was the recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON CYBERNETICS.



Yun Li (S'87–M'90) received the B.Sc. degree in electronics science from Sichuan University, Chengdu, China, the M.Sc. degree in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, and the Ph.D. degree in computing and control engineering from the University of Strathclyde, Glasgow, U.K., in 1984, 1987, and 1990, respectively.

From 1989 to 1990, he was with U.K. National Engineering Laboratory and Industrial Systems and Control Ltd., Glasgow, U.K. He joined the University of Glasgow, Glasgow, as a Lecturer in 1991, where he is currently a Professor of Systems Engineering. He served as a Founding Director of the University of Glasgow Singapore, Singapore, from 2011 to 2013 and acted as a Founding Director of the university's international joint program with UESTC, Chengdu, in 2013. He established the IEEE Computer-Aided Control System Design Evolutionary Computation Working Group and the European Network of Excellence in Evolutionary Computing Workgroup on Systems, Control, and Drives in 1998. He was invited to Kumamoto University, Kumamoto, Japan, as a Visiting Professor in 2002 and is currently a Visiting Professor to UESTC and Sun Yat-sen University, Guangzhou, China. He has authored the interactive online courseware EA Demo for *Evolutionary Computation* and has over 180 publications.

Dr. Li is a Chartered Engineer.